RC25566 (WAT1510-065) October 20, 2015
Mathematics

IBM Research Report

On Some Spanning Tree Games

Mourad Baïou
CNRS and Université Clermont II
Campus des Cézeaux BP 125
63173 Aubière Cedex
France

Francisco Barahona
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

ON SOME SPANNING TREE GAMES

MOURAD BAÏOU AND FRANCISCO BARAHONA

Abstract. We investigate four games related to spanning trees in a graph. First we study a two-person zero-sum game where an evader every few minutes chooses a spanning tree in a network and sends a signal. Then an inspector also every few minutes chooses an edge to inspect. The evader needs a tree selection strategy that minimizes the average probability of being detected. The inspector seeks to maximize this probability. We show that finding the evader's optimal strategy reduces to finding a maximum packing of spanning trees, and finding the inspector's strategy reduces to computing the strength of a graph. Then we treat another game where an attacker chooses a set of edges to destroy, and an inspector chooses an edge to find the attacker. We show that finding the inspector's strategy reduces to finding a minimum spanning tree, and the attacker strategy is given by a maximum packing of partitions of the node-set.

Then we study two cooperative games in a graph. In the first game each coalition corresponds to a set S of edges, and the value of it is the maximum number of disjoint spanning trees included in S. In the second game each coalition also corresponds to a set of edges S, and the value of the coalition is the number of new connected components created after removing the edges in S. For these two games we show that testing whether the core is non-empty, finding an element of the core, and testing membership to the core can be done in polynomial time.

1. Introduction

We consider some two-person zero-sum games and some cooperative games related to spanning trees in a graph. Our work on two-person games has been motivated by the following two-person game studied in [22]. Each day an "evader" selects a path from a node s to a node t in a network. Also each day an "inspector" selects an arc a in the network, and sets up an inspection site there. If the evader traverses arc a, he is detected with probability p_a, otherwise he goes undetected. The inspector has to find a probabilistic arc-inspection strategy which maximizes the average probability of detecting the evader, called the *interdiction probability*. The evader has to find a path-selection strategy which minimizes the interdiction probability. It is shown in [22] that finding the evader's strategy reduces to a maximum flow problem, and finding the inspector's strategy reduces to a minimum cut problem. In our case we assume that the evader every few minutes picks a spanning tree in a graph and sends a signal to all nodes. Then the inspector every few minutes chooses an edge e and inspects it. If edge e is being used to send the signal, this is detected with probability p_e. The inspector has to find a probabilistic edge-inspection strategy which maximizes the average probability of detecting the signal. The evader has to find a tree-selection strategy that minimizes the probability of being detected. Here we show that the evader's strategy is given by a maximum packing of spanning trees, while the inspector's strategy is obtained by computing the strength of a graph. Then we study a second two-person game where an

Date: October 16, 2015.
Key words and phrases. spanning trees, two-person games, strength of a network, spanning trees.

attacker chooses a set of edges to disconnect a network, then an inspector chooses edges trying to detect the attacker. Here attacker's optimal strategy is given by a maximum packing of partitions of the node-set, and the inspector's strategy is given by a minimum weighted spanning tree.

We also study two cooperative games in a graph. In the first game each coalition corresponds to a set S of edges, and the value of it is the maximum number of disjoint spanning trees included in S. In the second game each coalition also corresponds to a set of edges S, and the value of the coalition is the number of new components created after removing the edges in S. For these two games we study the algorithmic aspects of testing whether the core is non-empty, finding an element of the core, and testing membership to the core. The core is one of the most important concepts in cooperative game theory. Other cooperative games related to combinatorial optimization have been studied in [18], [16], [15], [9], [8], [7], [10], [14], see also [6] and the references therein.

This paper is organized as follows. In Section 2 we define some notation, and review some polyhedral results related to spanning trees. In Section 3 we study two-person games, and in Section 4 we treat cooperative games.

2. Preliminaries

In this section we define some notation and review some results from polyhedral combinatorics related to spanning trees.

2.1. **Notation.** Consider a graph $G = (V, E)$, for a partition $\{S_1, \ldots, S_p\}$ of V, let $\delta(S_1, \ldots, S_p)$ be the set of edges with both endpoints in different sets of the partition. Throughout this paper we assume that $p \geq 2$ each time that we refer to a partition $\{S_1, \ldots, S_p\}$ of V. We use n to denote $|V|$ and m to denote $|E|$.

For $F \subseteq E$ the *incidence vector* of F, $x^F : E \to \Re$, is defined by $x^F(e) = 1$ if $e \in F$, and $x^F(e) = 0$ otherwise. For a function $w : E \to \Re$ and $S \subseteq E$, we use $w(S)$ to denote $\sum_{e \in S} w(e)$.

2.2. **The strength problem.** Given a graph $G = (V, E)$ and a capacity function $w : E \to \mathbb{Z}_+$, a solution of the integer program

$$\max \sum_T y_T \tag{1}$$

$$\sum \{y_T \mid e \in T\} \leq w(e) \text{ for each edge } e \in E \tag{2}$$

$$y_T \geq 0, \text{ integer valued, for each spanning tree } T, \tag{3}$$

is called an *integral packing of spanning trees*. A min-max relation for (1)-(3) was given by Tutte [21] and Nash-Williams [19], as follows.

Theorem 1. *A graph $G = (V, E)$ has k disjoint spanning trees if and only if for every partition $\{S_1, \ldots, S_p\}$ of V,*

$$|\delta(S_1, \ldots, S_p)| \geq k(p-1).$$

Notice that if an edge e has a integer nonnegative capacity $w(e)$, we can make $w(e)$ parallel copies of the edge e, then this theorem implies that the value of the maximum in (1)-(3) is

$$\min \left\lfloor \frac{w(\delta(S_1, \ldots, S_p))}{p-1} \right\rfloor, \tag{4}$$

where the minimum is taken among all partitions of V, and $\lfloor z \rfloor$ denotes the largest integer less than or equal to z. Polynomial algorithms for finding a maximum integral (and fractional) packing of spanning trees have been given in [3] and [12].

If we relax the integrality condition in (3), and we only ask for variables y to be nonnegative, we obtain a linear program. It also follows from Theorem 1 that the optimal value is

$$\min \frac{w(\delta(S_1, \ldots, S_p))}{p-1}, \tag{5}$$

where the minimum is taken among all partitions of V. The value of this minimum was called the *strength of the network* in [5]. The dual problem is

$$\min wx \tag{6}$$
$$x(T) \geq 1, \text{ for each spanning tree } T, \tag{7}$$
$$x \geq 0. \tag{8}$$

This implies that the extreme points of the polyhedron defined by (7)-(8) are among the vectors that are obtained as follows: for each partition $\{S_1, \ldots, S_p\}$ of V, take the incidence vector of $\delta(S_1, \ldots, S_p)$ and divide it by $(p-1)$. However one should notice that not all vectors obtained in this way are extreme points. For instance, assume that the graph consists of just one tree, then the extreme points are obtained only when $p = 2$.

Cunningham [5] gave an algorithm for finding the minimum in (5) and (4), it requires $O(nm)$ minimum cut problems. Later an algorithm that requires n applications of the preflow algorithm was given in [4].

2.3. **Minimum spanning trees.** Here we use some results of the theory of Blocking polyhedra [11]. Consider the polyhedron defined by (7)-(8), its Blocking polyhedron is defined as follows. Let B be a matrix whose columns correspond to the edges, and whose rows are the incidence vectors of $\delta(S_1, \ldots, S_p)$, divided by $p-1$, for each partition $\{S_1, \ldots, S_p\}$ of V. Thus the rows of B contain all extreme points of the polyhedron defined by (7)-(8). Then the Blocking polyhedron is defined by the system of inequalities below.

$$Bx \geq 1, \tag{9}$$
$$x \geq 0. \tag{10}$$

It follows from the theory of Blocking polyhedra that its extreme points are the incidence vectors of all spanning trees of G. Thus the linear program

$$\min dx \tag{11}$$
$$\text{subject to (9)-(10)}, \tag{12}$$

can be solved with Kruskal's algorithm [17] for minimum spanning trees. Now we discuss how to solve the dual problem

$$\max y1 \tag{13}$$
$$yB \leq d, \tag{14}$$
$$y \geq 0. \tag{15}$$

We call this is a maximum packing of partitions.

First we consider the linear program below where inequalities (9) have been replaced by an equivalent form.

$$\min dx \tag{16}$$

subject to

$$x\big(\delta(S_1,\dots,S_p)\big) \geq (p-1), \text{ for all partitions } \Phi = \{S_1,\dots,S_p\} \text{ of } V, \tag{17}$$

$$x(e) \geq 0. \tag{18}$$

The dual is

$$\begin{aligned} &\max \sum \gamma_\Phi \,(p_\Phi - 1)\\ &\sum_{\{\Phi:\, e\in\delta(S_1,\dots,S_p)\}} \gamma_\Phi \leq d(e), \quad \text{for each edge } e,\\ &\gamma \geq 0. \end{aligned}$$

Here for each partition $\Phi = \{S_1,\dots,S_p\}$ we define $p_\Phi = p$. The algorithm below is a primal-dual version of Kruskal's algorithm.

Algorithm A

- **Step 0**. Start with $\bar\gamma = 0$, $\bar x = 0$, $\bar d(e) = d(e)$ for all $e \in E$. Also start with the partition Φ consisting of all singletons.
- **Step 1**. Compute $\bar\epsilon = \min\{\bar d(e) : e \in \delta(S_1,\dots,S_p)\}$. Update $\bar\gamma_\Phi \leftarrow \bar\gamma_\Phi + \bar\epsilon$, $\bar d(e) \leftarrow \bar d(e) - \bar\epsilon$ for all $e \in \delta(S_1,\dots,S_p)$. Here $\Phi = \{S_1,\dots,S_p\}$.
- **Step 2**. Let $\bar e$ be an edge giving the value of $\bar\epsilon$, in Step 1. Set $\bar x(\bar e) = 1$. Let S_i and S_j the sets in Φ containing the endnodes of $\bar e$. Set $\Phi' = \Phi \setminus \{S_i, S_j\} \cup \{S_i \cup S_j\}$, this means shrink S_i and S_j into one set.
- **Step 3**. If $\Phi' = \{V\}$ stop. Otherwise set $\Phi \leftarrow \Phi'$ and go to Step 1.

This algorithm performs exactly $n-1$ iterations. It is easy to see that $\bar\gamma$ is feasible at every step, and if the edge weights d are integer valued, then $\bar\gamma$ and $\bar d$ are integer valued at every step. At the end $\bar x$ is the incidence vector of a spanning tree. It is also easy to see that at the end the pair $(\bar x, \bar\gamma)$ satisfy the complementary slackness conditions. Thus $(\bar x, \bar\gamma)$ are optimal solutions. Let $\bar y$ be defined as $\bar y_\Phi = \bar\gamma_\Phi(p_\Phi - 1)$ for each partition $\Phi = \{S_1,\dots,S_p\}$ of V. Then $\bar y$ is a solution of (13)-(15). Notice that if d is integer valued, then $\bar\gamma$ and $\bar y$ are also integer valued.

For a family of inequalities $\mathcal{F}$, the *separation problem* consists of given a vector $\bar x$, find an inequality in $\mathcal{F}$ that is violated by $\bar x$ if there is one, or prove that none exists. For inequalities (17), an algorithm for the separation problem was given in [5], it requires m minimum cut problems. Later an algorithm requiring n minimum cut problems was given in [2]. This will be used in Section 4.

3. Two-person games

In this section we study two two-person zero-sum games related to spanning trees.

3.1. **The network strength game.** Consider an undirected graph $G = (V, E)$, and suppose that every few minutes a player (the evader) picks a spanning tree and sends a signal to all nodes. Then a second player (the inspector) every few minutes chooses an edge e and inspects it. If edge e is being used to send the signal, this is detected with probability p_e. The inspector has to find a probabilistic edge-inspection strategy which maximizes the average probability of detecting the signal. The evader has to find a tree-selection strategy that minimizes the average probability of being detected. Here we show that the inspection strategy is obtained by computing the strength of an associated graph, while the evader's strategy is given by a maximum packing of spanning trees in the same graph. The special case when $p_e = 1$ for each edge e, has been treated in [1] and called the "Wiretap game." The inspector's strategy was studied in [1], but finding the evader's strategy was left as an open question.

Let y_T be the probability that the evader chooses the tree T. For the inspector, let x_e be the probability of choosing the edge e. If the inspector chooses an edge e, there is a probability p_e of detecting the signal if this is at this edge. Let P be a diagonal matrix that contains the probabilities $\{p_e\}$, and x a row vector, then xP is a row with components $\{x_e p_e\}$. Let D be a matrix whose rows correspond to the edges, and whose columns are the incidence vectors of all trees. Let y be a column vector whose components are $\{y_T\}$. Then Dy is a column whose component associated with an edge e is the probability that the signal can be found at edge e. Thus $xPDy$ is the probability that the signal will be detected. We have to concentrate on the following two-person game:

$$\max_x \min_y \; xPDy \tag{19}$$

$$\sum_e x_e = 1, \tag{20}$$

$$\sum_T y_T = 1, \tag{21}$$

$$x \geq 0, \tag{22}$$

$$y \geq 0. \tag{23}$$

A reference on this type of games is [20]. If we fix y we have

$$\begin{aligned} &\max_x \; xPDy \\ &\sum_e x_e = 1, \\ &x \geq 0. \end{aligned}$$

And its dual is

$$\begin{aligned} &\min_\mu \mu \\ &\mu \geq p_e \sum \{y_T \,|\, e \in T\}, \quad \text{for each edge } e. \end{aligned}$$

Then (19)-(23) is equivalent to

$$\begin{aligned} &\min_{\mu,y} \; \mu \\ &\mu - p_e \sum \{y_T \,|\, e \in T\} \geq 0, \quad \text{for each edge } e, \\ &\sum_T y_T = 1, \\ &y \geq 0. \end{aligned}$$

This can be written as

$$\min_{\mu,y} \ \mu \tag{24}$$

$$\sum\{y_T \,|\, e \in T\} \leq \frac{\mu}{p_e}, \quad \text{for each edge } e, \tag{25}$$

$$\sum y_T = 1, \tag{26}$$

$$y \geq 0. \tag{27}$$

Here we are looking for the minimum value of μ such that there is a packing of spanning trees of value one, and edge capacities $\{\mu/p_e\}$.

Consider now the following linear program

$$\max \sum_T y_T \tag{28}$$

$$\sum\{y_T \,|\, e \in T\} \leq \frac{1}{p_e}, \quad \text{for each edge } e, \tag{29}$$

$$y \geq 0. \tag{30}$$

Here we are looking for a maximum packing of spanning trees, with edge capacities $\{1/p_e\}$. This was discussed in Sub-section 2.2. Let $\{\hat{y}_T\}$ be a solution of (28)-(30). Let $\lambda = \sum_T \hat{y}_T$, then if we set

$$\mu = \frac{1}{\lambda},$$

and $\bar{y}_T = \frac{1}{\lambda}\hat{y}_T$ for each spanning tree T, we obtain a solution of (24)-(27).

Thus at this point we have found the value of the game (19)-(23), this is μ, and the values for the variables y. Now we have to find the values of the variables x. Let a partition $\{S_1, \ldots, S_q\}$ be a solution of the strength problem (5), with edge capacities $\{1/p_e\}$. This was treated in Sub-section 2.2. Recall that λ is the value of a solution of the strength problem. Then we set

$$\bar{x}_e = \begin{cases} \frac{1}{\lambda p_e(q-1)} & \text{if } e \in \delta(S_1, \ldots, S_q), \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

Thus

$$\sum_{e \in \delta(S_1,\ldots,S_q)} \bar{x}_e = \frac{1}{\lambda} \sum_{e \in \delta(S_1,\ldots,S_q)} \frac{1}{p_e(q-1)} = \frac{1}{\lambda}\lambda = 1.$$

The complementary slackness conditions for the linear program (6)-(8) and its dual imply that for $e \in \delta(S_1, \ldots, S_q)$,

$$\sum\{\hat{y}_T \,|\, e \in T\} = \frac{1}{p_e},$$

and

$$\sum\{\bar{y}_T \,|\, e \in T\} = \frac{1}{\lambda p_e}.$$

Then

$$\bar{x}PD\bar{y} = \sum_{e \in \delta(S_1,\ldots,S_q)} p_e \frac{1}{\lambda p_e(q-1)} \frac{1}{\lambda p_e} = \frac{1}{\lambda} = \mu. \tag{32}$$

This shows that $(\bar{x}, \bar{y})$ are solutions of (19)-(23). Recall that algorithms for packing spanning trees have been given in [3] and [12], and algorithms for the strength problem appear in [5] and [4]. Thus we can state the main result of this section.

Theorem 2. *Optimal strategies for both players can be computed in polynomial time. The evader strategy can be obtained from a maximum packing of spanning trees with edge capacities $\{1/p_e\}$. The inspector strategy can be obtained from a solution of the strength problem, also with edge capacities $\{1/p_e\}$.*

3.2. **The network attack game.** Let $G = (V, E)$ be an undirected graph. Suppose that each day, an attacker chooses a set of edges to destroy and disconnect the network. Each day, an inspector chooses an edge e, and if the attacker is at this edge it will be intercepted with probability p_e. Let D be a matrix whose rows correspond to the edges and whose columns correspond to edge sets, so that an element associated with an edge e and a set S is $1/k(S)$ if $e \in S$, and 0 otherwise. Here we denote by $k(S)$ the number of new connected components obtained after removing the edges in S. Let P be a diagonal matrix whose elements are p_e for each edge e. We study the two-person zero-sum game

$$\max_x \ \min_y \ xPDy \tag{33}$$

$$\sum_e x_e = 1, \tag{34}$$

$$\sum_S y_S = 1, \tag{35}$$

$$x \geq 0, \tag{36}$$

$$y \geq 0. \tag{37}$$

Here

$$xPDy = \sum_S \frac{y_S}{k(S)} \sum_{e \in S} x(e)p(e).$$

So for a given inspection strategy x, the attacker chooses an edge set S that minimizes $\sum_{e \in S} x(e)p(e)/k(S)$. This is the average detection probability per new component created. We should notice that the attacker should only use edge sets of the form

$$T = \delta(S_1, \ldots, S_p), \tag{38}$$

where $\{S_1, \ldots, S_p\}$ is a partition of V. To see this notice that if $k(T) \geq 1$, then there is a partition $\{S_1, \ldots, S_p\}$ of V with $\delta(S_1, \ldots, S_p) \subseteq T$ and $p - 1 = k(T)$. So in what follows we assume that all columns of D correspond to edge-sets like in (38).

If we fix y we have

$$\max_x \ xPDy$$
$$\sum_e x_e = 1,$$
$$x \geq 0.$$

And its dual is

$$\min_\mu \mu$$
$$\mu \geq p_e \sum \{y_T/k(T) \ : \ e \in T\}, \quad \text{for each edge } e.$$

Then (33)-(37) is equivalent to

$$\min_{\mu,y} \ \mu$$
$$\mu - p_e \sum\{y_T/k(T) \,:\, e \in T\} \geq 0, \quad \text{for each edge } e,$$
$$\sum_T y_T = 1,$$
$$y \geq 0.$$

This can be written as

$$\min_{\mu,y} \ \mu \tag{39}$$
$$\sum\{y_T/k(T) \,:\, e \in T\} \leq \frac{\mu}{p_e}, \quad \text{for each edge } e, \tag{40}$$
$$\sum y_T = 1, \tag{41}$$
$$y \geq 0. \tag{42}$$

Here we are looking for the minimum value of μ such that there is a packing of partitions of value one, and edge capacities $\{\mu/p_e\}$.

Consider now the following linear program

$$\max \sum_T y_T \tag{43}$$
$$\sum\{y_T/k(T) \,:\, e \in T\} \leq \frac{1}{p_e}, \quad \text{for each edge } e, \tag{44}$$
$$y \geq 0. \tag{45}$$

This is problem (13)-(15) in Sub-section 2.3, so we are looking for a maximum packing of partitions, with edge capacities $\{1/p_e\}$. Let $\{\hat{y}_T\}$ be a solution of (43)-(45). Let $\lambda = \sum_T \hat{y}_T$, then if we set

$$\mu = \frac{1}{\lambda},$$

and $\bar{y}_T = \frac{1}{\lambda}\hat{y}_T$ for each edge-set T, we obtain a solution of (39)-(42).

Thus we have the value of the game (33)-(37), this is μ, and we have the values of the variables y. Now we have to find the value of the variables x. As seen in Sub-section 2.3, a solution of the dual of (43)-(45) is a minimum weighted spanning tree $\mathscr{T}$, with edge weights $\{1/p_e\}$.

Recall that λ is the weight of $\mathscr{T}$. We set

$$\bar{x}_e = \begin{cases} \frac{1}{\lambda p_e} & \text{if } e \in \mathscr{T}, \\ 0 & \text{otherwise.} \end{cases} \tag{46}$$

Then

$$\sum_{e \in \mathscr{T}} \bar{x}_e = \frac{1}{\lambda} \sum_{e \in \mathscr{T}} \frac{1}{p_e} = \frac{1}{\lambda}\lambda = 1.$$

The complementary slackness conditions for (11)-(12) and its dual imply that for $e \in \mathscr{T}$,

$$\sum\{\hat{y}_T/k(T) \,:\, e \in T\} = \frac{1}{p_e},$$

and

$$\sum\{\bar{y}_T/k(T)\,:\,e\in T\}=\frac{1}{\lambda p_e}.$$

Then

$$\bar{x}PD\bar{y}=\sum_{e\in\mathscr{T}}p_e\frac{1}{\lambda p_e}\sum\{\bar{y}_T/k(T)\,:\,e\in T\}=\sum_{e\in\mathscr{T}}p_e\frac{1}{\lambda p_e}\frac{1}{\lambda p_e}=\frac{1}{\lambda}=\mu. \tag{47}$$

This shows that $(\bar{x},\bar{y})$ are solutions of (19)-(23). Recall that $\bar{x}$ can be obtained with Kruskal's algorithm [17], and $\bar{y}$ can be obtained with the algorithm of Sub-section 2.3. Thus we can state the main result of this Sub-section.

Theorem 3. *Optimal strategies for both players can be computed in polynomial time. The attacker strategy can be obtained from a maximum packing of partitions with edge capacities $\{1/p_e\}$. The inspector strategy can be obtained from a minimum weighted spanning tree, also with weights $\{1/p_e\}$.*

4. Cooperative Games

Now we use similar techniques to investigate two cooperative games.

4.1. The tree-packing game.

First we study a cooperative game in an undirected graph $G=(V,E)$. Each player is associated with an edge $e\in E$, and we have a value function $v:2^E\to\Re$. For each coalition of players $S\subseteq E$, $v(S)$ is the revenue that the subset S of players can obtain by forming a coalition of the players in S only. In the *tree-packing game* (E,v), for each coalition $S\subseteq E$, $v(S)$ is the maximum number of disjoint spanning trees included in S.

The income distributed to individual players is a vector $x:E\to\Re_+$, satisfying $x(E)=v(E)$. Next we study the core of this game.

4.1.1. *The core.*

This is a concept introduced by Gillies [13]. It is based on the following stability condition: no subgroup of players will do better if they break away from the joint decision of all players to form their own coalition. The *core* of the game is the following polyhedron

$$C(v)=\{x\in\Re^E\,|\,x(E)=v(E),\ x(S)\ge v(S),\ \text{for } S\subseteq E\}.$$

Next we show how to test if the core is nonempty, how to find an element of the core, and how to decide if a vector is in the core. The core of other combinatorial optimization games has been studied in [18], [16], [15], [9], [8], [7], [10], [14], and others. We need the following lemma.

Lemma 4. *For $S\subseteq E$, the inequality $x(S)\ge v(S)$ is implied by*

$$x(T)\ge 1,\ \textit{for each spanning tree } T\subseteq S,$$
$$x(e)\ge 0,\ \textit{for each edge } e\in S.$$

Proof. If S does not contain a spanning tree, then $x(S)\ge 0$ is implied by $x(e)\ge 0$, for each edge $e\in S$.

If S contains a spanning tree consider the inequalities

$$x(T)\ge 1,\ \text{for each spanning tree } T\subseteq S,$$
$$x(e)\ge 0,\ \text{for each edge } e\in S.$$

As seen in Sub-section 2.2, this system implies $x(S) \geq f(S)$, where $f(S)$ is the value of a maximum fractional packing of spanning trees included in S. Since the value of a fractional packing is at least the value of an integral packing we have $f(S) \geq v(S)$, and $x(S) \geq v(S)$. □

This lemma shows that the core can be written as

$$x(E) = v(E) \tag{48}$$

$$x(T) \geq 1, \text{ for each spanning tree } T \subseteq E, \tag{49}$$

$$x(e) \geq 0, \text{ for each edge } e \in E. \tag{50}$$

We also have that $x(E) \geq f(E)$, where $f(E)$ is the value of a maximum fractional packing of spanning trees. Since $f(E) \geq v(E)$, we have that the core is nonempty if and only if $f(E) = v(E)$. This and the results of Sub-section 2.2 imply the following.

Theorem 5. *The core is nonempty if and only if*

$$\min \frac{|\delta(S_1, \ldots, S_p)|}{p-1} \tag{51}$$

is an integer number. This minimum is taken among all partitions $\{S_1, \ldots, S_p\}$ of V.

Theorem 6. *We can test whether the core is nonempty in polynomial time.*

Given a vector $\bar{x}$, testing if $\bar{x}$ satisfies inequalities (49) reduces to a minimum weight spanning tree problem. Thus we have the following.

Theorem 7. *Given a vector $\bar{x}$, we can decide in polynomial time if $\bar{x}$ belongs to the core.*

As seen in Sub-section 2.2, we can also describe the core as the convex hull of a set of vectors as follows.

Theorem 8. *Let λ be the value of the minimum in (51). If the core is nonempty, it is the convex hull of all vectors obtained as follows:*

- *Pick a partition $\{S_1, \ldots, S_p\}$ with*
$$|\delta(S_1, \ldots, S_p)| = \lambda(p-1).$$
- *Let $\bar{x}$ be the incidence vector of this partition. Pick $\bar{x}/(p-1)$.*

Since we can find a solution of (51) in polynomial time, we have the following.

Theorem 9. *If the core is non-empty, we can find a member of the core in polynomial time.*

4.1.2. *A relaxation of the core.* If the core is empty, the *multiplicative ϵ-core* is a relaxation proposed in [9]. Given $\epsilon \in [0, 1]$, it is defined by the following set of inequalities.

$$x(E) = v(E) \tag{52}$$

$$x(S) \geq (1-\epsilon)\, v(S), \text{ for } S \subset E. \tag{53}$$

We need the following lemma.

Lemma 10. *For $S \subseteq E$, the inequality $x(S) \geq (1-\epsilon)v(S)$ is implied by*

$$x(T) \geq 1-\epsilon, \text{ for each spanning tree } T \subset S,$$

$$x(e) \geq 0, \text{ for each edge } e \in S.$$

Proof. If $v(S) = 0$, then $x(S) \geq 0$ is implied by $x(e) \geq 0$, for all $e \in S$.

If $v(S) \geq 1$, let $k = v(S)$. There are k disjoint spanning trees $T_1, \ldots, T_k$ included in S. Then

$$x(S) \geq x(T_1) + \ldots + x(T_k) \geq k(1-\epsilon) = v(S)(1-\epsilon).$$

□

Then we have the following system.

$$\begin{aligned} &x(E) = v(E) \\ &x(T) \geq 1-\epsilon, \text{ for each spanning tree } T, \\ &x(e) \geq 0, \text{ for each edge } e \in E. \end{aligned}$$

We can look for the minimum value of ϵ so that this set is not empty as follows. Define $x' = x/(1-\epsilon)$, then we have

$$\begin{aligned} &x'(E) = v(E)/(1-\epsilon) \\ &x'(T) \geq 1 \text{ for each spanning tree } T, \\ &x'(e) \geq 0, \text{ for each edge } e \in E. \end{aligned}$$

Thus the minimum value of ϵ is obtained minimizing $x'(E)$ subject to

$$\begin{aligned} &x'(T) \geq 1 \text{ for each spanning tree } T, \\ &x'(e) \geq 0, \text{ for each edge } e \in E. \end{aligned}$$

This reduces to the strength problem as seen in Sub-section 2.2. Thus we can state the following.

Theorem 11. *The smallest value of ϵ so that the multiplicative ϵ-core is non-empty, can be found in polynomial time.*

4.2. **The network disconnection game.** For a connected graph $G = (V, E)$, this is a cooperative game (E, v), where for each coalition $S \subseteq E$, $v(S)$ is the number of new connected components obtained after removing the edges in S. Before giving a system of inequalities that defines the core we need the lemma below whose proof is omitted.

Lemma 12. *In the definition of the core we only need inequalities $x(T) \geq v(T)$ when $T = \delta(S_1, \ldots, S_p)$, for each partition $\{S_1, \ldots, S_p\}$ of V, and $x(e) \geq 0$ for each edge e.*

Thus the core is defined by

$$\begin{aligned} &x(E) = n-1, \\ &x(\delta(S_1, \ldots, S_p)) \geq p-1, \text{ for all partitions } \{S_1, \ldots, S_p\} \text{ of } V, \\ &x(e) \geq 0. \end{aligned} \tag{54}$$

As seen in Sub-section 2.3, this is the spanning tree polytope, i.e., the convex hull of incidence vectors of spanning trees of G. Since the graph is connected, the core is non-empty, and the incidence vector of any spanning tree is an element of the core. The separation problem for inequalities (54) can be solved in polynomial time with the algorithms of [5] and [2]. We summarize all this in the following theorem.

Theorem 13. *The following problems can be solved in polynomial time:*

- *Finding an element of the core.*
- *Deciding if a given vector belongs to the core.*

References

[1] H. Aziz, O. Lachish, M. Paterson, and R. Savani, *Wiretapping a hidden network*, in Internet and Network Economics, Springer, 2009, pp. 438–446.

[2] F. Barahona, *Separating from the dominant of the spanning tree polytope*, Operations research letters, 12 (1992), pp. 201–203.

[3] ———, *Packing spanning trees*, Mathematics of Operations Research, 20 (1995), pp. 104–115.

[4] E. Cheng and W. H. Cunningham, *A faster algorithm for computing the strength of a network*, Information Processing Letters, 49 (1994), pp. 209–212.

[5] W. H. Cunningham, *Optimal attack and reinforcement of a network*, Journal of the ACM (JACM), 32 (1985), pp. 549–561.

[6] X. Deng and Q. Fang, *Algorithmic cooperative game theory*, in Pareto Optimality, Game Theory And Equilibria, Springer, 2008, pp. 159–185.

[7] X. Deng, T. Ibaraki, and H. Nagamochi, *Algorithmic aspects of the core of combinatorial optimization games*, Mathematics of Operations Research, 24 (1999), pp. 751–766.

[8] X. Deng and C. H. Papadimitriou, *On the complexity of cooperative solution concepts*, Mathematics of Operations Research, 19 (1994), pp. 257–266.

[9] U. Faigle and W. Kern, *On some approximately balanced combinatorial cooperative games*, Zeitschrift für Operations Research, 38 (1993), pp. 141–152.

[10] U. Faigle, W. Kern, S. P. Fekete, and W. Hochstättler, *On the complexity of testing membership in the core of min-cost spanning tree games*, International Journal of Game Theory, 26 (1997), pp. 361–366.

[11] D. R. Fulkerson, *Blocking and anti-blocking pairs of polyhedra*, Mathematical programming, 1 (1971), pp. 168–194.

[12] H. N. Gabow and K. Manu, *Packing algorithms for arborescences (and spanning trees) in capacitated graphs*, Mathematical Programming, 82 (1998), pp. 83–109.

[13] D. B. Gillies, *Solutions to general non-zero-sum games*, Contributions to the Theory of Games, 4 (1959), pp. 47–85.

[14] M. X. Goemans and M. Skutella, *Cooperative facility location games*, Journal of Algorithms, 50 (2004), pp. 194–214.

[15] D. Granot and F. Granot, *Computational complexity of a cost allocation approach to a fixed cost spanning forest problem*, Mathematics of Operations Research, 17 (1992), pp. 765–780.

[16] D. Granot and G. Huberman, *On the core and nucleolus of minimum cost spanning tree games*, Mathematical programming, 29 (1984), pp. 323–347.

[17] J. B. Kruskal, *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proceedings of the American Mathematical society, 7 (1956), pp. 48–50.

[18] N. Megiddo, *Computational complexity of the game theory approach to cost allocation for a tree*, Mathematics of Operations Research, 3 (1978), pp. 189–196.

[19] C. S. J. Nash-Williams, *Edge-disjoint spanning trees of finite graphs*, Journal of the London Mathematical Society, 1 (1961), pp. 445–450.

[20] G. Owen, *Game theory*, Academic Press, 1995.

[21] W. T. Tutte, *On the problem of decomposing a graph into n connected factors*, Journal of the London Mathematical Society, 1 (1961), pp. 221–230.

[22] A. Washburn and K. Wood, *Two-person zero-sum games for network interdiction*, Operations Research, 43 (1995), pp. 243–251.

(M. Baïou) CNRS, and Université Clermont II, Campus des cézeaux BP 125, 63173 Aubière cedex, France.

E-mail address, M. Baïou: baiou@isima.fr

(F. Barahona) IBM T. J. Watson research Center, Yorktown Heights, NY 10589, USA.

E-mail address, F. Barahona: barahon@us.ibm.com